What is Coopr?

Coopr is a COmmon Optimization Python Repository. Essentially, its a collection of packages for Python that aid in solving optimization problems such as linear programming.

Coopr implementations tend to come in two parts: the model and the data file. The purpose of the model is to create an abstract representation of the problem while the data file is the concrete inputs unique to each situation. By splitting the problem into two parts, a single model can be used for a myriad of different situations. For example, in the classic diet problem we only need to construct one model and it can be used to solve the problem for if we restrict the foods to just McDonalds food, Chinese food, or even if we use all foods ever (though inputting data for all food ever is not recommended). Similarly, we can change the nutrients and their requirements based on the needs of each person. In fact, this method even lets us create a working model before all the data is gathered, saving time and generating results faster.

The model

The model is where we define our important sets, parameters, variables and rules. These documents can be thought of as a rough, abstract representation of the problem on hand. The model itself should not contain any data, but it should contain the sets and structures that will contain the data.

When creating the model, its important to think about what are the main items being investigated (for example, factories and roads), values associated with them (supply, demand, cost to move supplies), the values were trying to maximize or minimize (usually cost), and different constraints that must be taken into account (the amount shipped should exceed the demand, but be less than the supply). All of these will be represented in the model.

The model file is saved with a .py extension and might have a name such as "DietModel.py".

The data file

The data file is where all the specific numbers related to an individual problem are defined. Here, the sets and parameters are filled in with information to be used for the problem. For example, if we were solving the diet problem, the data file is where we input the foods and nutrients we are concerned with. This is the concrete representation of the problem to be solved.

Typically the data file is one of the last parts of the model that's written and the one that's most likely to change. While one model file can be used to solve many different problems, the data file used will always be different.

The data file is saved with a .dat extension and is usually named something along the lines of "DietData1.dat".